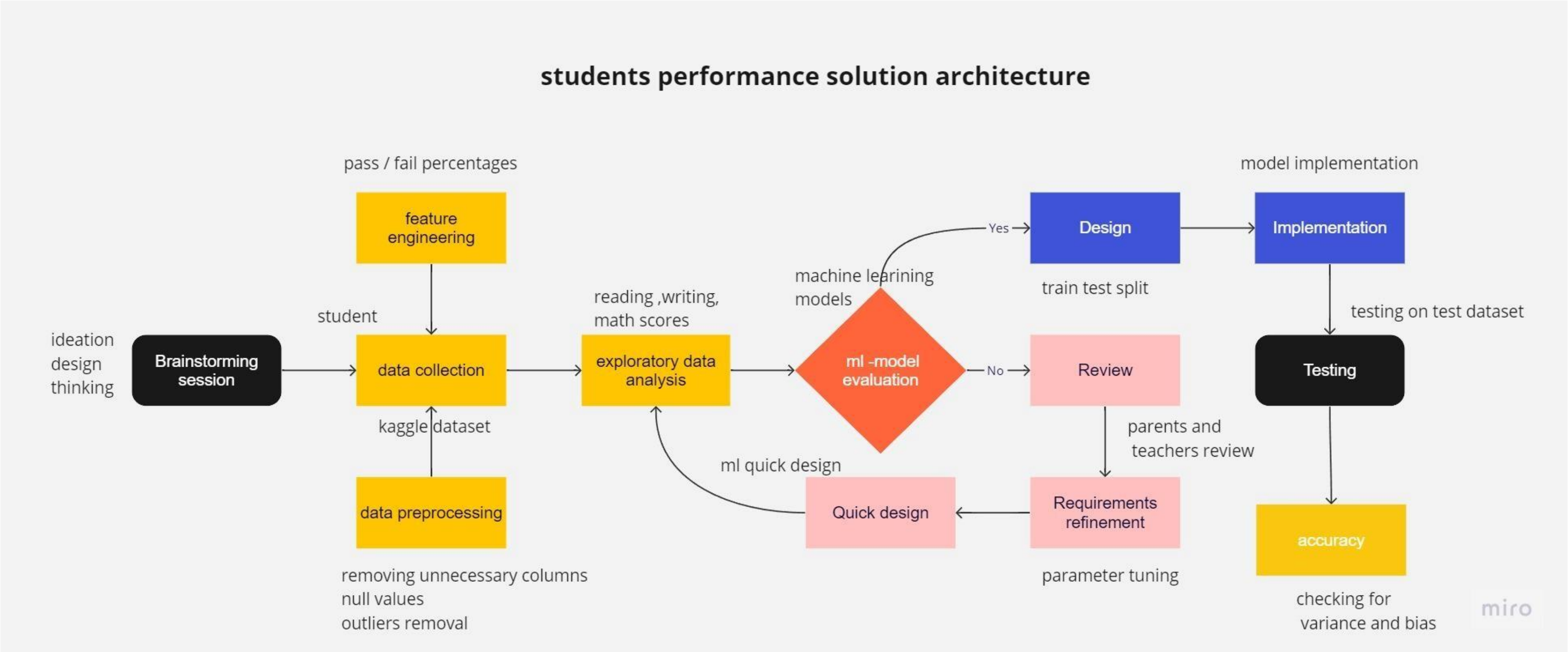
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	19 oct 2023
Team ID	B59ACB2EE24AC56B74C4C89695327FAA
Project Name	Unleashing the potential of the youth: student performance analysis
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



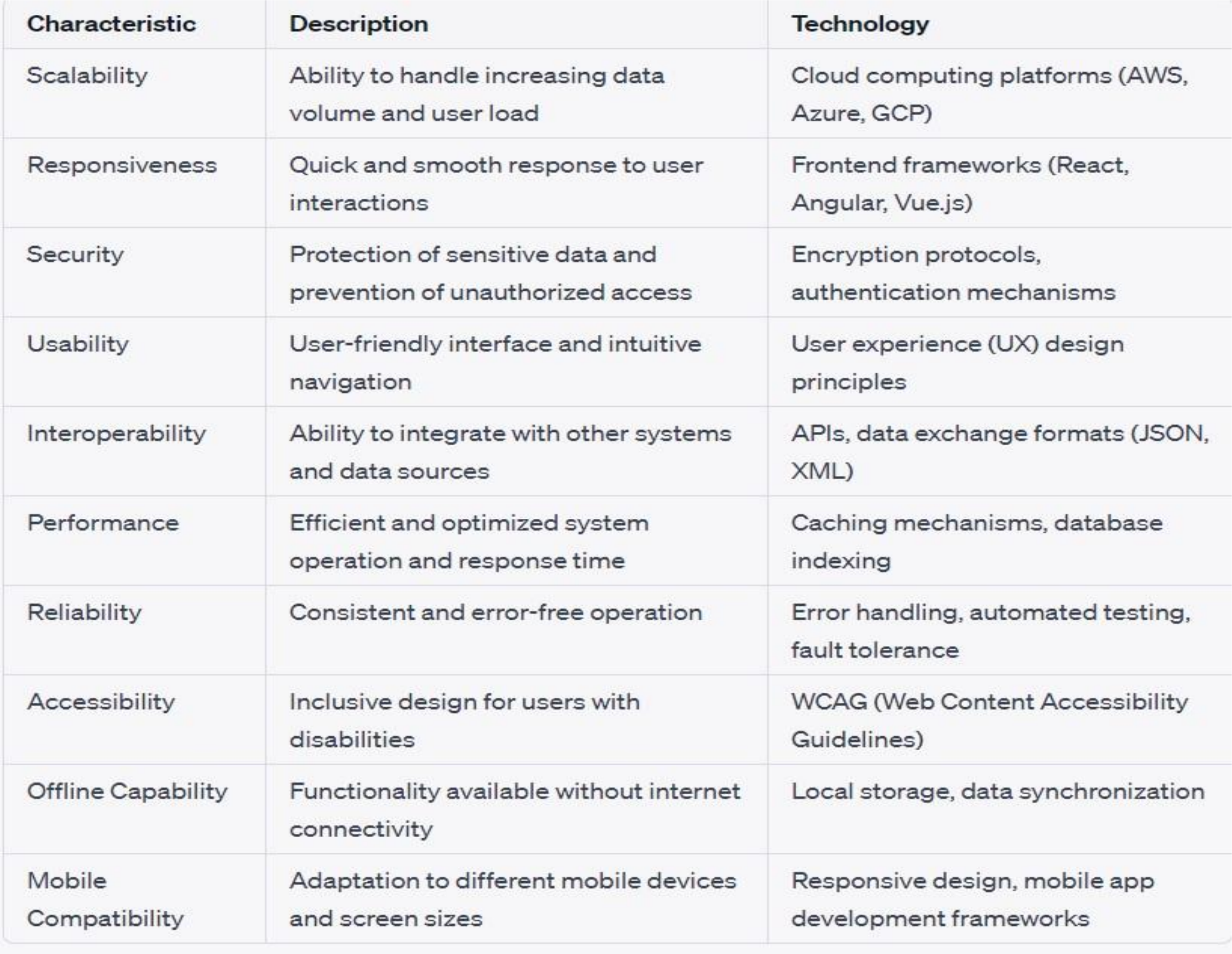
students performance solution architecture

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interface component provides the visual and interactive elements of the system that users interact with.	HTML, CSS, JavaScript

2.	Application Logic-1	Logic for a process in the application	Python
3.	Database	The DBMS component is responsible for storing and managing the system's data, including student records, grades, and performance metrics.	MySQL,
4.	Cloud Database	Cloud services provide scalable infrastructure, storage, and computing resources for the system. They can also offer additional services like authentication, messaging, or data processing	IBM DB2,
5.	File Storage	File storage requirements	Local Filesystem
6.	External API-1	External API used in the application	IBM Weather API, etc.
7.	Machine Learning Model	This component handles data processing, analysis, and machine learning tasks, such as generating performance metrics, predicting future performance, or identifying patterns in student data	Object Recognition Model, etc.
8.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:



Characteristic	Description	Technology
Scalability	Ability to handle increasing data volume and user load	Cloud computing platforms (AWS, Azure, GCP)
Responsiveness	Quick and smooth response to user interactions	Frontend frameworks (React, Angular, Vue.js)
Security	Protection of sensitive data and prevention of unauthorized access	Encryption protocols, authentication mechanisms
Usability	User-friendly interface and intuitive navigation	User experience (UX) design principles
Interoperability	Ability to integrate with other systems and data sources	APIs, data exchange formats (JSON, XML)
Performance	Efficient and optimized system operation and response time	Caching mechanisms, database indexing
Reliability	Consistent and error-free operation	Error handling, automated testing, fault tolerance
Accessibility	Inclusive design for users with disabilities	WCAG (Web Content Accessibility Guidelines)
Offline Capability	Functionality available without internet connectivity	Local storage, data synchronization
Mobile Compatibility	Adaptation to different mobile devices and screen sizes	Responsive design, mobile app development frameworks

References:

1. Research Paper: "Predicting Student Performance: An Application of Data Mining Techniques" by Romil Bhardwaj and Priya Soni.
2. Article: "Non-Functional Requirements: What Are They and Why Are They Important?" by Rachel Thomas, available at: https://reqtest.com/requirements-blog/non-functional-requirements-important/
3. Article: "Functional Requirements vs Non-Functional Requirements" by Swati Seela, available at: https://reqtest.com/requirements-blog/functional-requirements-vs-non-functional-requirements/
4. Article: "Building Modern Web Applications: A Comprehensive Guide" by Martin Fowler, available at: https://martinfowler.com/articles/web.html
5. Documentation: Official documentation and resources for technology frameworks and tools such as Django

   (https://docs.djangoproject.com/), React (https://reactjs.org/), Angular (https://angular.io/), Vue.js https://vuejs.org/(), MongoDB (https://docs.mongodb.com/) ıd AWS (https://aws.amazon.com/documentation/).